Would be great with proper scanning. You can do that with the Scannable mobile app.

Machine Learning over Networks

Problem -> 1.1

required.

A differentiable function f is μ strongly convex iff

$\forall x_1, x_2 \in \mathcal{X}$, $\mu > 0$

$$f(x_2) \geq f(x_1) + \nabla f(x_1)^T (x_2 - x_1) + \frac{\mu}{2} \|x_2 - x_1\|_2^2 \quad (1)$$

We can interchange x_2 and x_1 to write

$$f(x_1) \geq f(x_2) + \nabla f(x_2)^T (x_1 - x_2) + \frac{\mu}{2} \|x_2 - x_1\|_2^2 \quad (2)$$

Adding (1) & (2) we obtain:

$$f(x_1) + f(x_2) \geq f(x_1) + f(x_2) + \nabla f(x_1)^T (x_2 - x_1) - \nabla f(x_2)^T (x_2 - x_1) + \mu \|x_2 - x_1\|_2^2$$

$$\Rightarrow (\nabla f(x_2) - \nabla f(x_1))^T (x_2 - x_1) \geq \mu \|x_2 - x_1\|_2^2 \quad (3)$$

From inequality (3),

we can obtain:

$$\frac{1}{\mu} (\nabla f(x_2) - \nabla f(x_1))^T (x_2 - x_1) \geq \|x_2 - x_1\|_2^2$$

$$\Rightarrow \frac{1}{\mu} \|(\nabla f(x_2) - \nabla f(x_1))\|_2 \geq \frac{\|x_2 - x_1\|_2^2}{\|x_2 - x_1\|_2} \quad \text{(Taking the norm on both sides)}$$

$$\Rightarrow \frac{1}{\mu} \|\nabla f(x_2) - \nabla f(x_1)\|_2 \geq \|x_2 - x_1\|_2 \quad (4)$$

When a function f is just convex, the condition is

$$f(x_2) \geq f(x_1) + \nabla f(x_1)^T (x_2 - x_1)$$

for twice differentiable, $\nabla^2 f(x) \geq 0 \quad \forall x \in \mathcal{S}$

similarly for M strongly convex, where the condition is:

$$f(x_2) \geq f(x_1) + \nabla f(x_1)^T (x_2 - x_1) + \frac{M}{2} \|x_2 - x_1\|_2^2$$

we obtain: $\nabla^2 f(x) \geq M I_d$

Group 3: Would be nice if you annotated this section with a)

Group 3: Nice!

Group 3: Would be nice if you annotated this section with b).

Group 3: Would be nice if you annotated this section with c)

d.)

From (3), we have:

$$\|x_2 - x_1\|_2^2 \leq \frac{1}{M} (\nabla f(x_2) - \nabla f(x_1))^T (x_2 - x_1)$$

and from (4) we have:

$$\|x_2 - x_1\|_2 \leq \frac{1}{M} \|\nabla f(x_2) - \nabla f(x_1)\|_2$$

~~Replacing bound (4) in (3)~~

~~Squaring~~

~~[illegible]~~

Using the properties of L_2 norm, we have:

$$(\nabla f(x_2) - \nabla f(x_1))^T (x_2 - x_1) \leq \|\nabla f(x_2) - \nabla f(x_1)\|_2 \|x_2 - x_1\|_2 \quad (5)$$

Substituting $\|x_2 - x_1\|$'s bound from (4), in (5), we obtain:

$$(\nabla f(x_2) - \nabla f(x_1))^T (x_2 - x_1) \leq \|\nabla f(x_2) - \nabla f(x_1)\|_2 \cdot \frac{1}{M} \cdot \|\nabla f(x_2) - \nabla f(x_1)\|_2$$

$$\Rightarrow (\nabla f(x_2) - \nabla f(x_1))^T (x_2 - x_1) \leq \frac{1}{M} \|\nabla f(x_2) - \nabla f(x_1)\|_2^2$$

Group 3: Nice!

Group 3: Even better if you put a rectangle on this like you did with the others!

d.) To prove that $f(x) + r(x)$ is strongly for a convex f and a strongly convex r.

$$r(x_2) \geq r(x_1) + \nabla r(x_1)^T (x_2 - x_1) + \frac{M_1}{2} \|x_2 - x_1\|_2^2 \quad -(1)$$

$$f(x_2) \geq f(x_1) + \nabla f(x_1)^T (x_2 - x_1) + \frac{0 . M_2}{2} \|x_2 - x_1\|_2^2 \quad -(2)$$

Adding (1) & (2),

$$r(x_2) + f(x_2) \geq r(x_1) + f(x_1) + \nabla r(x_1)^T (x_2 - x_1) + \nabla f(x_1)^T (x_2 - x_1) + \frac{(M_1 + 0.M_2)}{2} \|x_2 - x_1\|_2^2$$

which is a $(M_1 + 0 \times M_2)$ strongly convex function.

Group 3: Where does 0 come from?

Group 3: Is this zero?

Group 3: What is 0

Would be nice to add "Given" here

P1.2. $\|\nabla f(x_2) - \nabla f(x_1)\|_2 \le L\|x_2 - x_1\|_2$ (1)

(a). Define $g(x) = \frac{L}{2}x^T x - f(x)$

$\nabla g(x) = Lx - \nabla f(x)$

(1) is equivalent to

$\langle Lx - \nabla f(x) - Ly + \nabla f(y), x - y\rangle \ge 0$

i.e., $\langle \nabla g(x) - \nabla g(y), x - y\rangle \ge 0$.

Then we have $g(x)$ is convex in x.

Thus. $g(y) \ge g(x) + \langle \nabla g(x), y - x\rangle$

Hence. $\frac{L}{2}y^T y - f(y) \ge \frac{L}{2}x^T x - f(x) + \langle Lx - \nabla f(x), y - x\rangle$

$\Rightarrow f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|x - y\|^2$ (2) complete.

(b) Define $z = y + \frac{1}{L}(\nabla f(x) - \nabla f(y))$

$f(y) - f(x) = f(y) - f(z) + f(z) - f(x)$

$\overset{(2)}{\ge} \langle \nabla f(y), z - y\rangle - \frac{L}{2}\|y - z\|^2 + \langle \nabla f(x), z - x\rangle$

$\overset{z.def}{\ge} \frac{1}{L}\langle \nabla f(y), \nabla f(x) - \nabla f(y)\rangle - \frac{L^2}{2L^2}\|\nabla f(x) - \nabla f(y)\|^2$

$+ \langle \nabla f(x), y - x\rangle + \langle \nabla f(x), \frac{1}{L}(\nabla f(x) - \nabla f(y))\rangle$

$= \langle \nabla f(x), y - x\rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2$. (3) complete.

(c) According to (3)

$f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2$

$f(x) \ge f(y) + \langle \nabla f(y), x - y\rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2$

Adding these two inequalities yields

$\langle \nabla f(x) - \nabla f(y), x - y\rangle \ge \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2$. complete

Group 3: Nice work!

1.3

Definition of convergence rates:

Suppose, $\lim_{n\to\infty} \frac{|X_{n+1} - X^*|}{|X_n - X^*|^q} = u$, where $u \in (0, \infty)$.

If $q = 1$, $\Rightarrow$ Linear convergence

If $q = 2$, $\Rightarrow$ quadratic convergence

If $q \in (1, 2)$ $\Rightarrow$ Superlinear convergence

Or

If $|X_{t+1} - X^*| \le c |X_t - X^*|$, $c \in (0, 1)$. $\Rightarrow$ Linear convergence

$(|X_T - X^*| \le c^T |X_0 - X^*|$

If $|X_{t+1} - X^*| \le c |X_t - X^*|^2$, $c \in (0, 1)$ $\Rightarrow$ Quadratic convergence.

If $|X_T - X^*| \le \frac{1}{T^k} |X_0 - X^*|$, $k > 1$ $\Rightarrow$ Sublinear convergence

Convergence rate.

Quadratic > superlinear > linear > sublinear.

Examples:

Gradient descent for smooth convex functions. $O(\frac{1}{\epsilon})$. $\Rightarrow$ sublinear

Gradient descent for smooth, strongly convex funcs. $O(\log \frac{1}{\epsilon}) \Rightarrow$ linear.

Newton method $\Rightarrow$ Quadratic $O(\log \log \frac{1}{\epsilon})$

Conjugate gradient for quadratic funcs $\Rightarrow$ superlinear.

Looks like you missed writing about the benefits of each convergence?

Group 3: Nice work!

Group 3: Is this 1?

1.5.

Prove: $(\nabla f(x) - \nabla f(y))^T (x-y) \geq \frac{mL}{m+L} \|x-y\|_2^2 + \frac{1}{m+L} \|\nabla f(x) - \nabla f(y)\|^2$

For strongly convex funcs, we have

$$\langle \nabla f(x) - \nabla f(y), x-y \rangle \geq m \|x-y\|^2$$

For L-smooth funcs, we have

$$\langle \nabla f(x) - \nabla f(y), x-y \rangle \geq \frac{1}{L} \|\nabla f(x) - \nabla f(y)\|^2$$

Define $g(x) = f(x) - \frac{m}{2}\|x\|^2$. $g(x)$ is convex since f is m-strongly convex.

$g(x)$ is also $(L-m)$-Lipschitz continuous.

Then, $\langle \nabla g(x) - \nabla g(y), x-y \rangle \geq \frac{1}{L-m} \|\nabla g(x) - \nabla g(y)\|^2$.

$$\langle \nabla f(x) - \nabla f(y) - m(x-y), x-y \rangle \geq \frac{1}{L-m} \|\nabla f(x) - \nabla f(y) - m(x-y)\|^2.$$

$$(1 + \frac{2m}{L-m}) \langle \nabla f(x) - \nabla f(y), x-y \rangle \geq \frac{1}{L-m} \|\nabla f(x) - \nabla f(y)\|^2 + (\frac{m^2}{L-m} + m) \|x-y\|^2.$$

The proof is complete by rearranging the terms.

Group 3: Would be great if you could show the rearrangements!

Problem 1.4

We have assumed strong convexity and smoothness on the function *f.*

Group 3: Would be great if you could add the problem definition here.

Assumption about constraint:

Nothing is said for the constraint **Ax=b,** which is a set of p linear constraints representing N-dimensional hyperplanes. We might need multiple assumptions. First, we assume that **A** is full row-rank (=p), thus no two hyperplanes are parallel. Second, we assume that p<N. Thus, the intersection is non-empty. Further, as intersection of convex regions is convex, the feasible region is also convex.

<u>Note about the constraint:</u>

During optimization, we can deal with the constraint potentially in different ways. If one uses the problem as it is, one always has to make sure that the solution at every iteration is indeed inside the feasibility region

One can choose to solve the dual problem, in case if that turns out to be easier and then the constraint vanishes and becomes $\lambda_i > 0$, which is the whole positive quadrant. This is particularly useful when A is tall, that is when N<p which is not the case in the given problem.

One can also convert the N dimensional constrained optimization problem to an (N-p) dimensional unconstrained optimization problem by rewriting the p variables in terms of N-p variables.

For example,

$$x_N=b_1-(A_{1,1}x_1+ A_{1,2}x_2+ A_{1,3}x_3+ \ldots + A_{1,N-1}x_{N-1}).$$

Now substitute x_{N-1} in terms of the second hyperplane in the constraint

Depending on the situation this might make the solution easy

(a)

With a small N, for example N=1000, one can easily use many standard algorithms for solving the problem. In such cases it is not difficult to find the gradient or hessian and thus can use whichever algorithm converges fast. **Gradient descent algorithm or Newtons method** are some examples of such algorithms.

(b)

In general, choice of an algorithm always depends of the tradeoff between number of iterations for sufficient convergence and the amount of computation per iterations. With a very large N the computation of derivatives and thus the best possible direction of the next-solution needs very large amount of calculations in each iteration. Thus, newtons method does not work well.

We thus suggest some randomized algorithms – like the **stochastic gradient descent**, or some deterministic sub-optimal but simple algorithms like the **coordinate descent** algorithm. In the former, one uses a direction which is randomized but is in the optimal half-space thus reducing the need of a lot of computations. In the latter, coordinate descent algorithm, one iteratively chooses one (or a set of) coordinate(s) our of the N coordinate and move in an optimum direction in the perspective of that (those) coordinate(s). This works because we have strict convexity.

(c)

Newtons method is not recommended for N=10^9. The reason is explained in the first paragraph of answer **(b).**

When the constraint is a probability simplex single hyperplane of the form $\mathbf{a}^T\mathbf{x}=b$, one might be able to approximate the hessian by ignoring many directions (that is coordinates) and concentrating on one or a subset of of coordinates chosen deterministically or randomly (recall stochastic gradient). Here it is easier to compute the next feasible solution point, because the number of second derivative computations are now reduced.

This method of course takes more iterations to converge but takes less computation per iterations. Further, due to the strong convexity, it is guaranteed to converge as we are decreasing the functional value in almost every steps (and some steps where the functional value is constant.)

This is also applicable to $1<p<<N$, just with potential changes in the convergence rate/complexity.

(d) The problem does not mention the convexity of the function **r(x)** but only mentions that it is twice differentiable. However, the selection of an algorithm depends precisely on this.
If r(x) is convex or strictly convex function, one can fall back to more or less the same answers given in parts **(a)** through **(c)**, even though the complexity slightly depends on the type of r(x).

One can still work with some descent algorithms with a different convergence rates if r(x) is quasi-convex or pseudo-convex. Descent algorithms will work in general as the global optimum is either unique, or otherwise at least a set of convex points, all of which giving the same functional value.

However, if r(x) is truly non-convex, one might has to settle for some general non-convex optimization techniques or some workarounds on the convex optimization techniques where one uses the properties of the particular r(x). For instance, for a function with two local minimas, one can find each of these two local minimas by intelligently choosing the starting point and then just compare them.

In overall, nice work!
(Where is the answer to problem 1.5 though?)